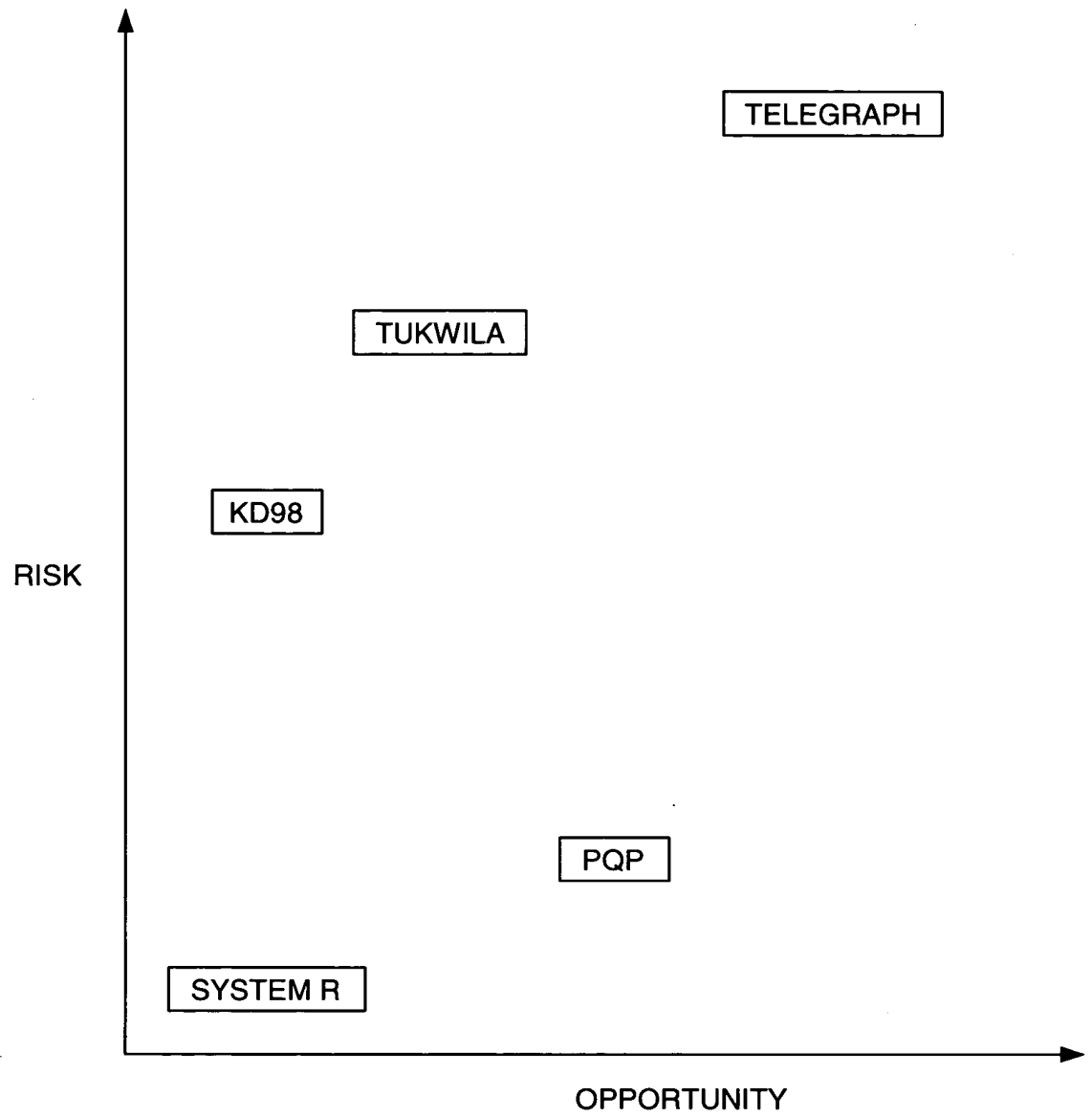
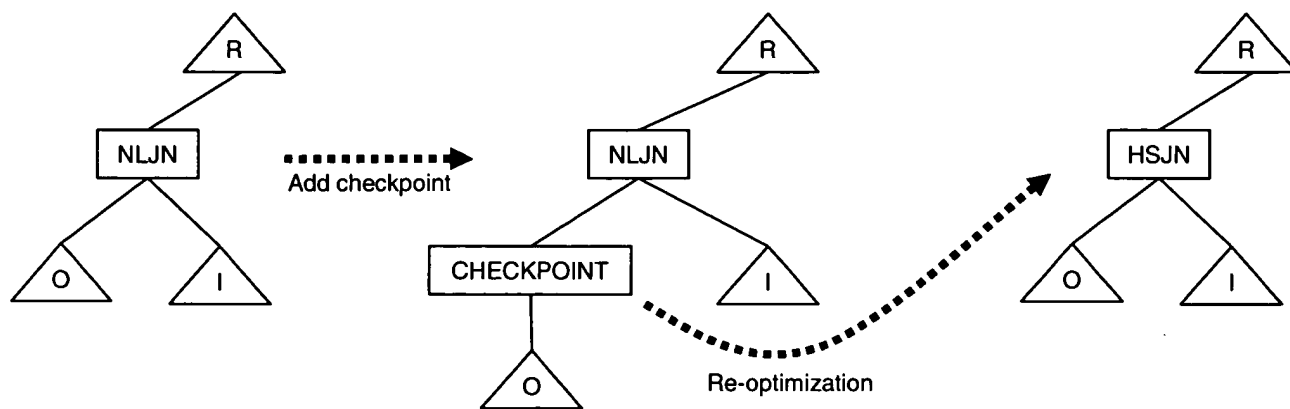


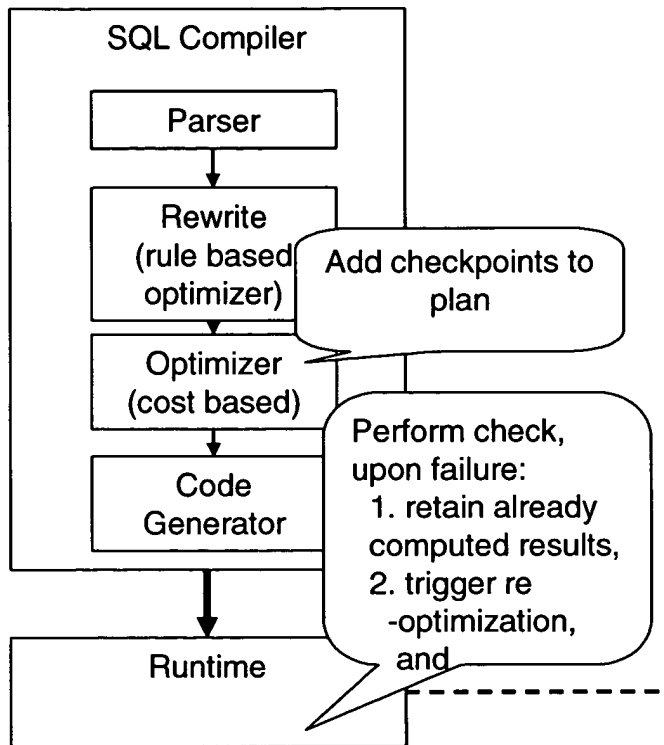
**FIG. 1**



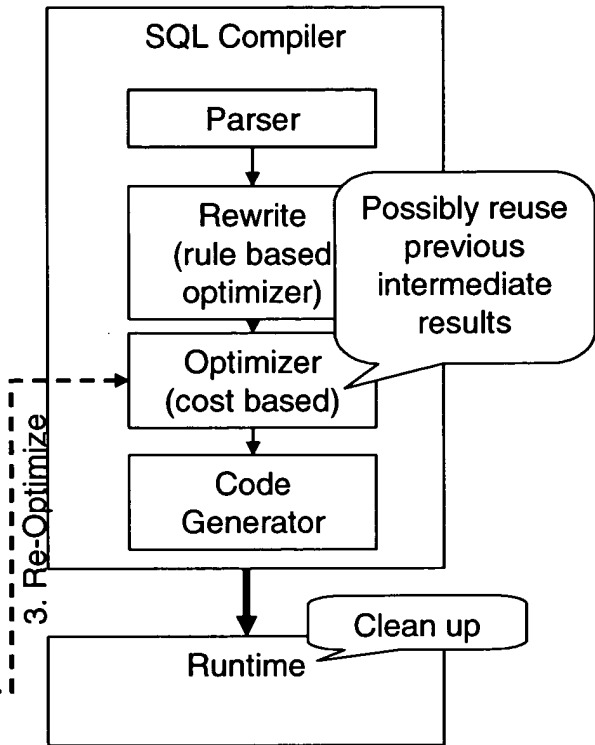


**FIG. 2**

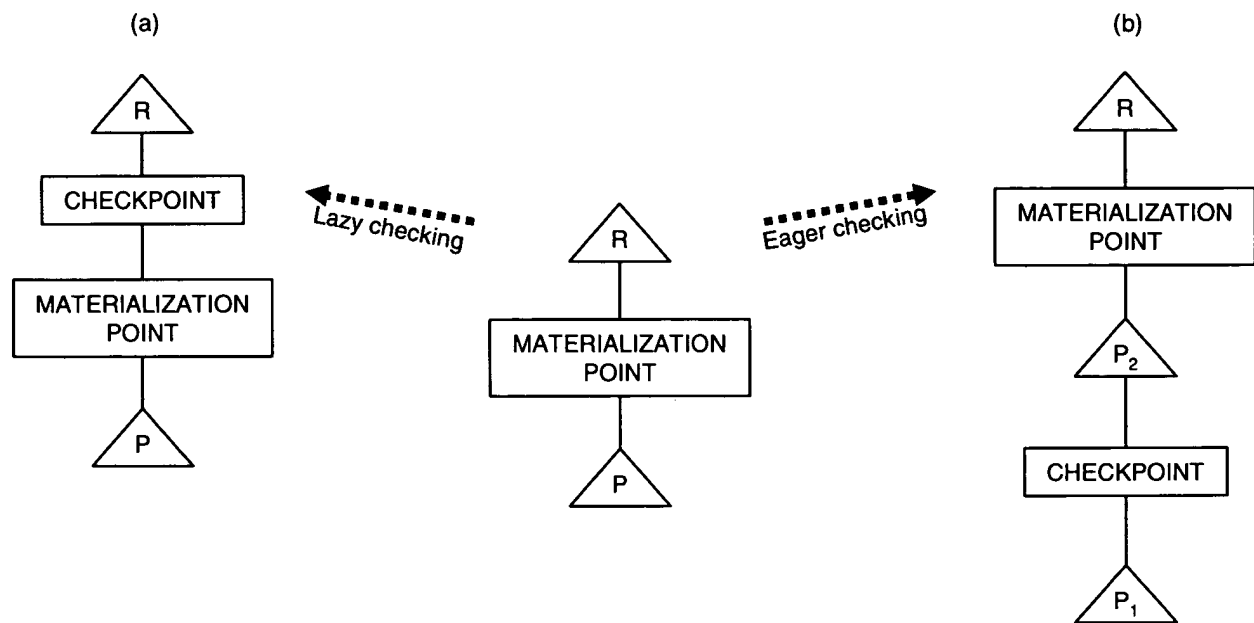
## I. Initial run



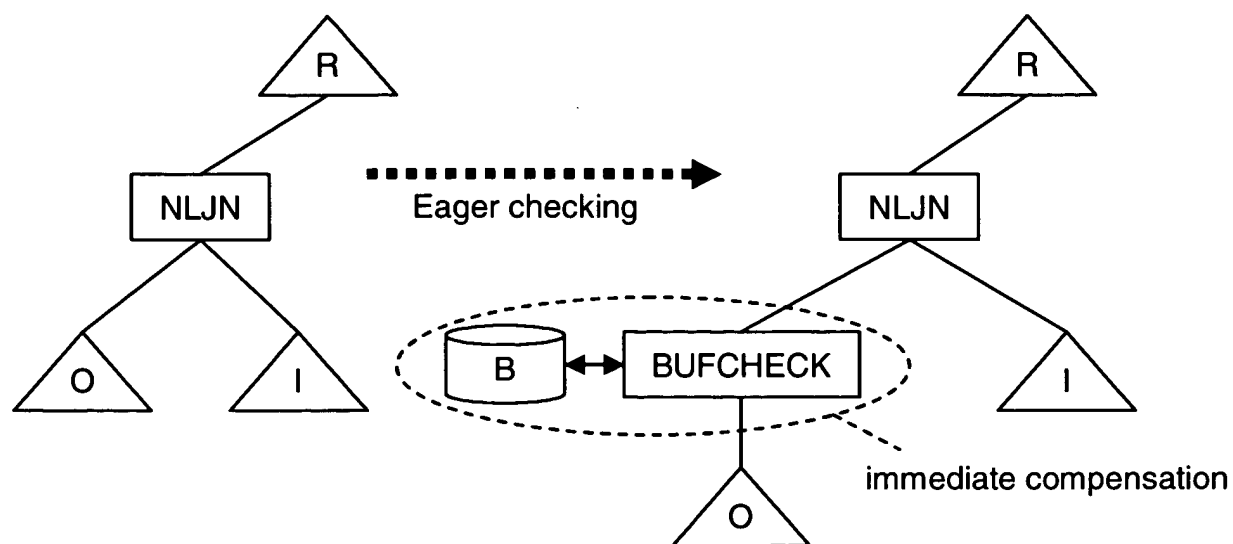
## II. Re-optimization



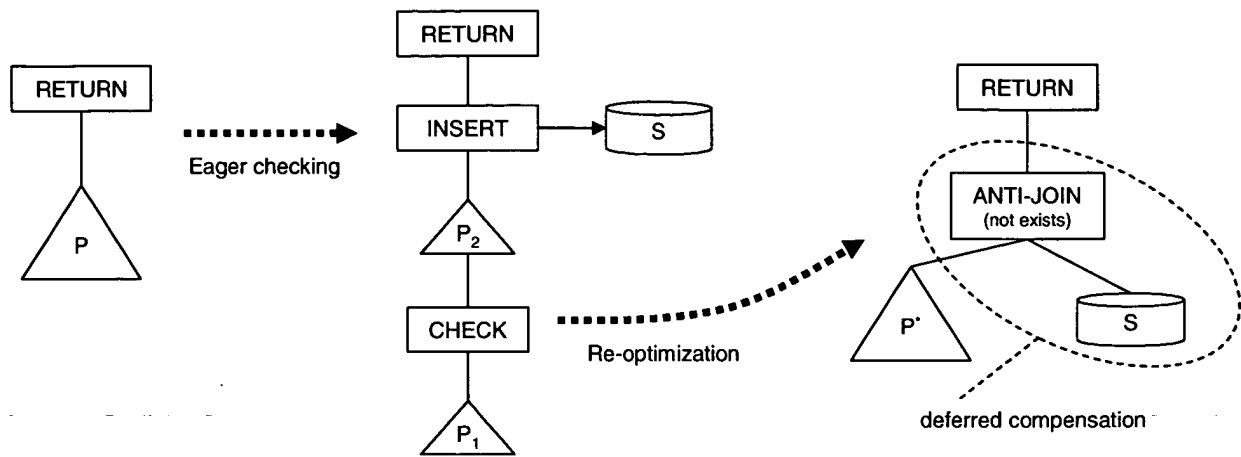
**FIG. 3**



**FIG. 4**



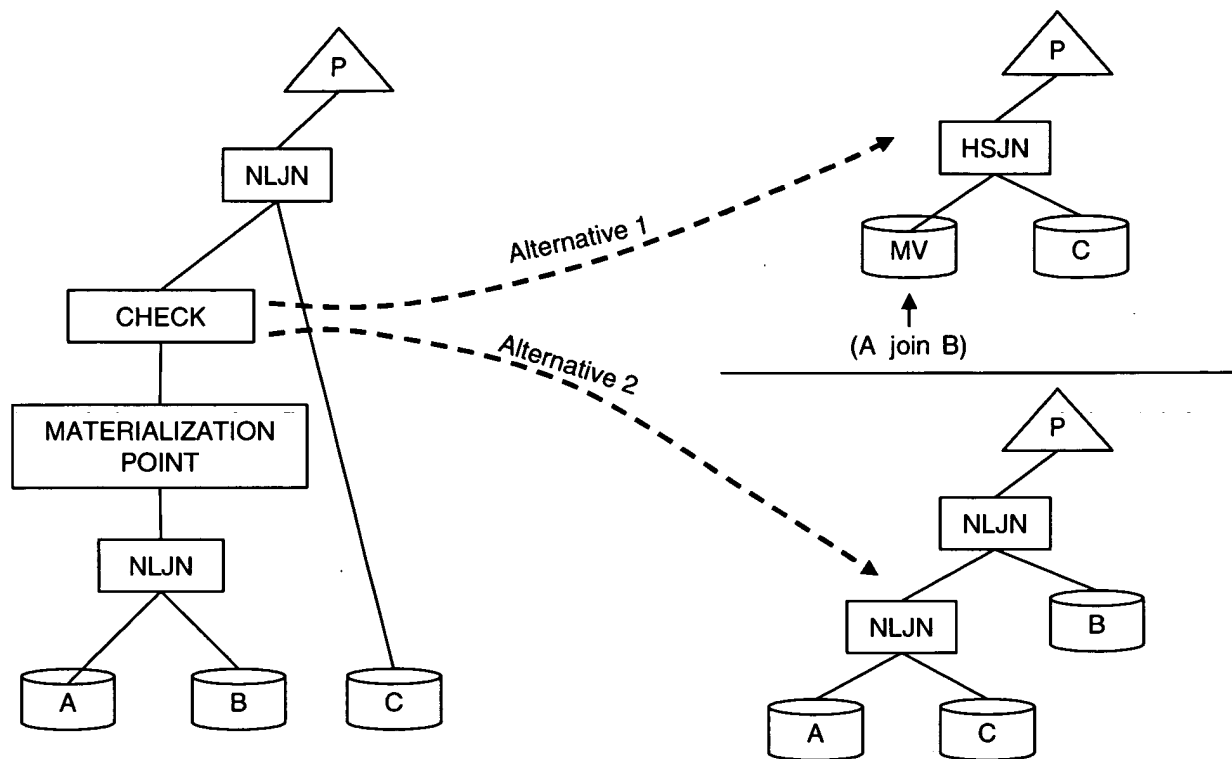
**FIG. 5**



**FIG. 6**

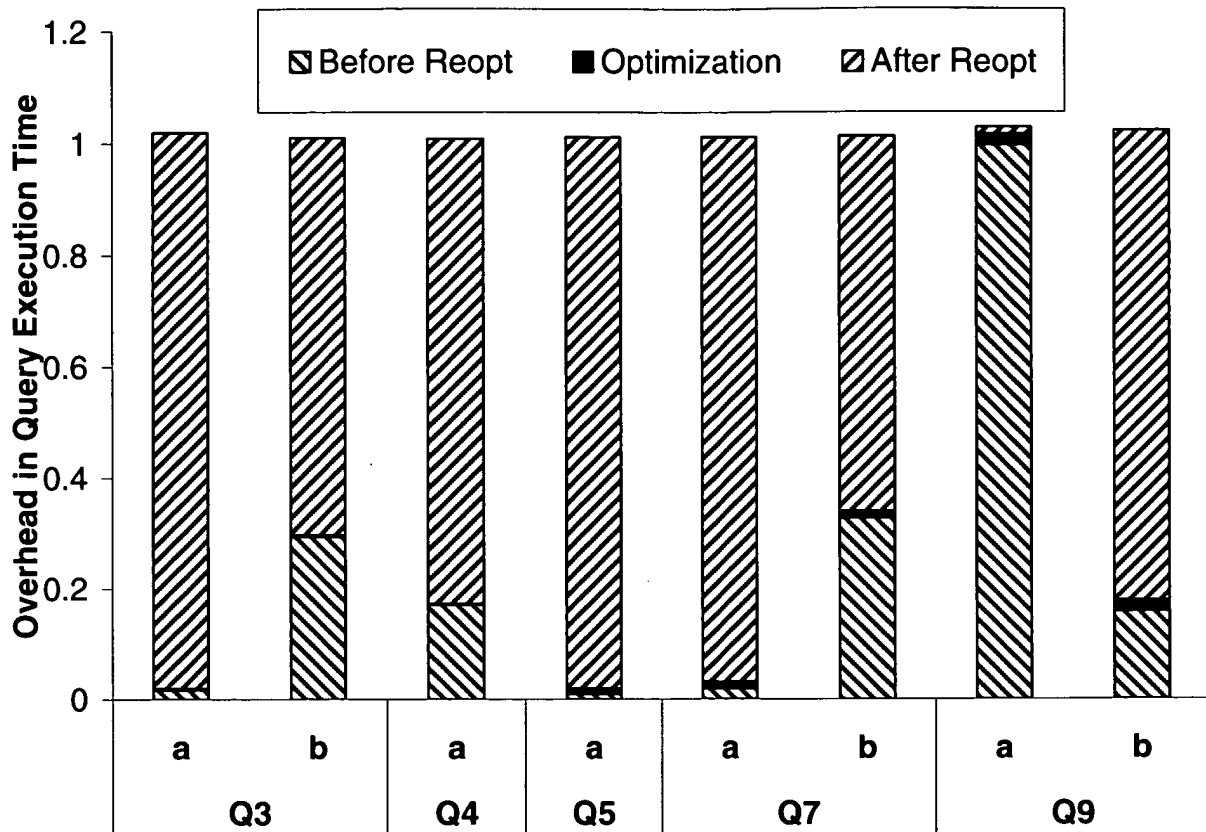
<p>CHECK.OPEN:</p> <pre> count = 0; </pre> <p>CHECK.NEXT:</p> <pre> count++; r = childStream.next(); if count &gt; high     call re-optimization; if count &lt; low and r = EOF     call re-optimization; else     return r; </pre> <p>CHECK.CLOSE:</p> <pre> Ø </pre>	<p>BUFCHECK.OPEN:</p> <pre> count = 0; allocate B of size b; // buffer for i = 0 to b do     B[i] = childStream.next();     if childStream.EOF()         and i &lt; low             call re-optimization; </pre> <p>BUFCHECK.NEXT:</p> <pre> count++; if high &lt; count     call re-optimization; if count &lt; b     return B[count]; else     return childStream.next(); </pre> <p>BUFCHECK.CLOSE:</p> <pre> free B; </pre>
--	--

**FIG. 7**

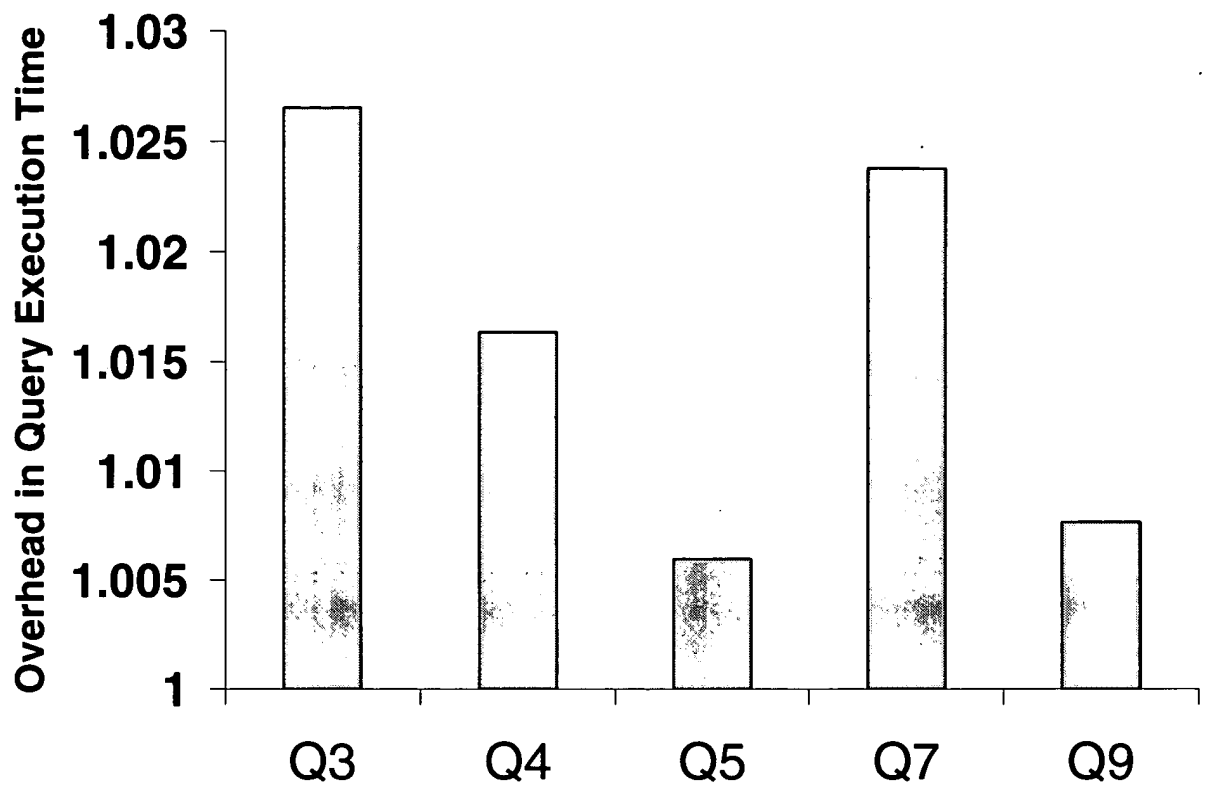


**FIG. 8**

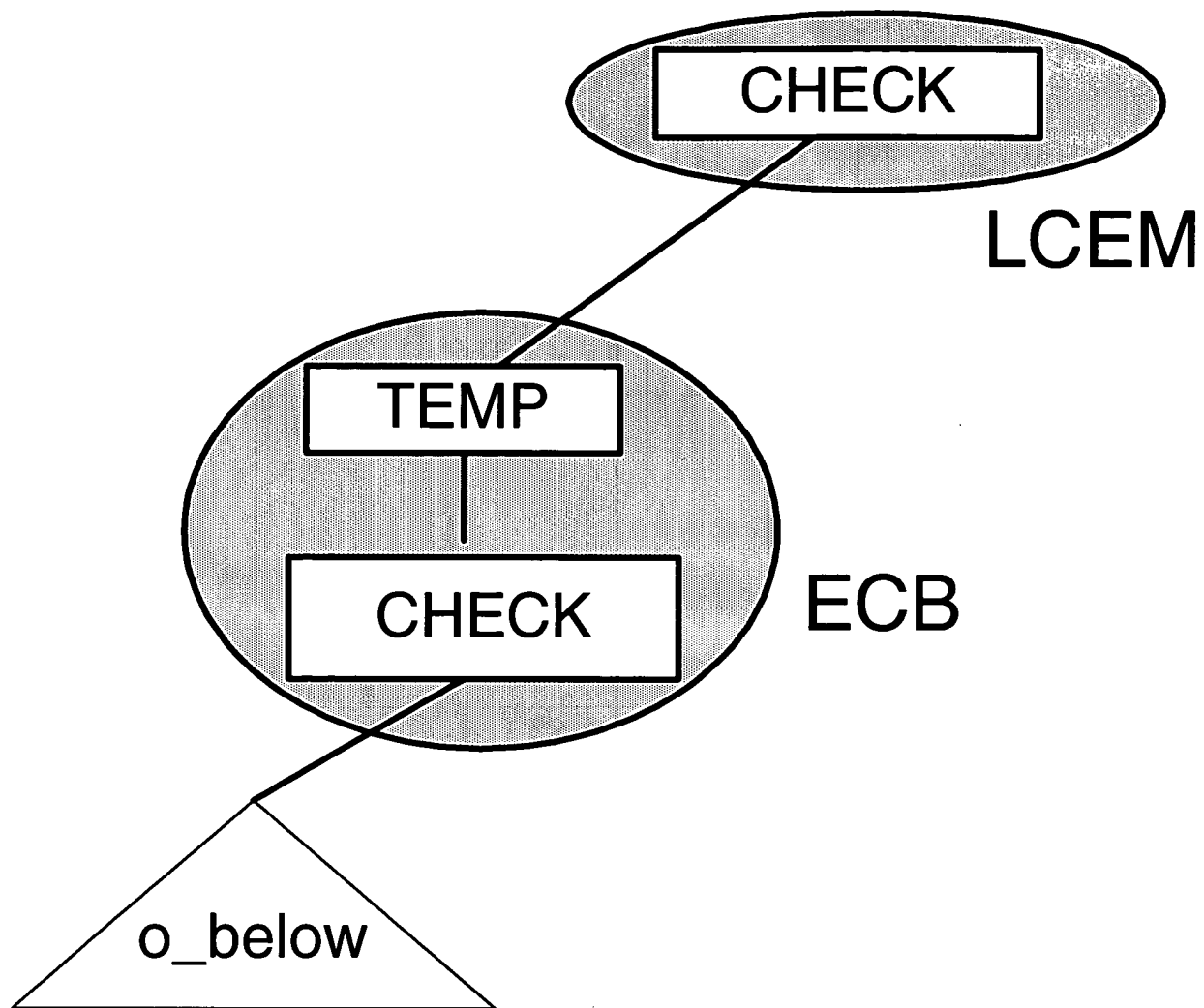




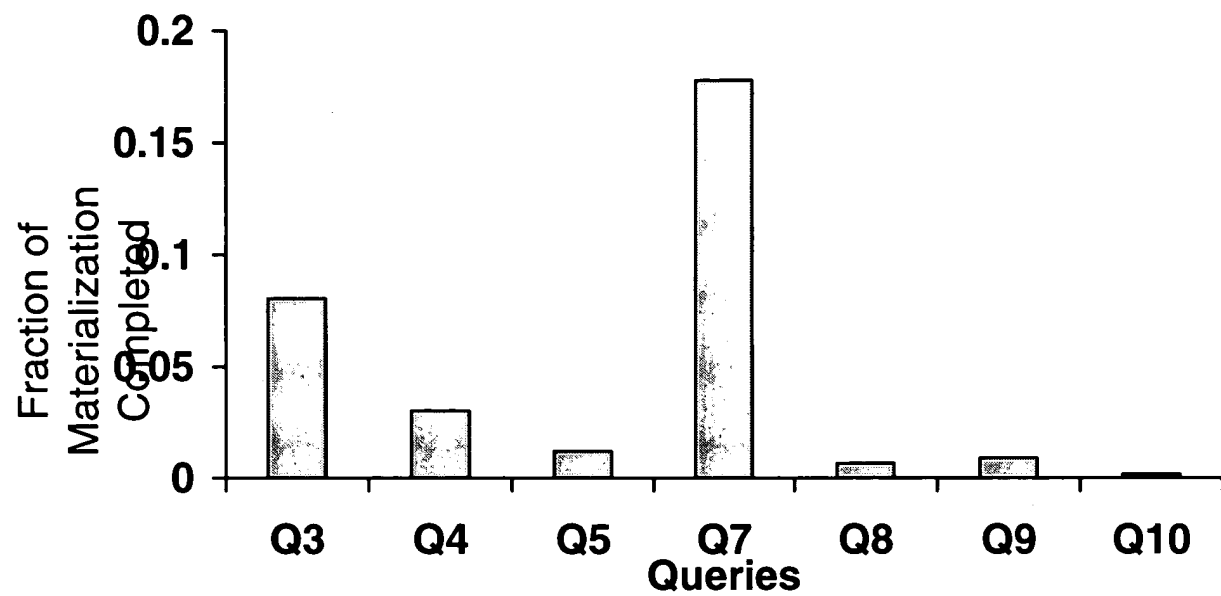
**FIG. 9**



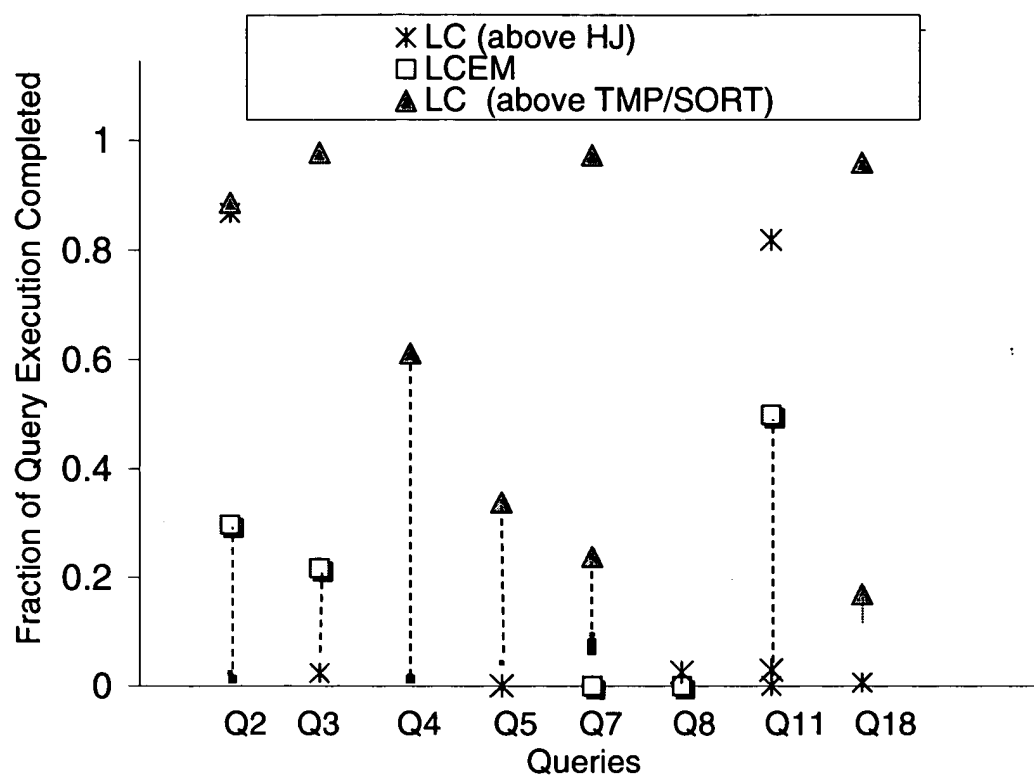
**FIG. 10**



**FIG. 11**



**FIG. 12**



**FIG. 13**

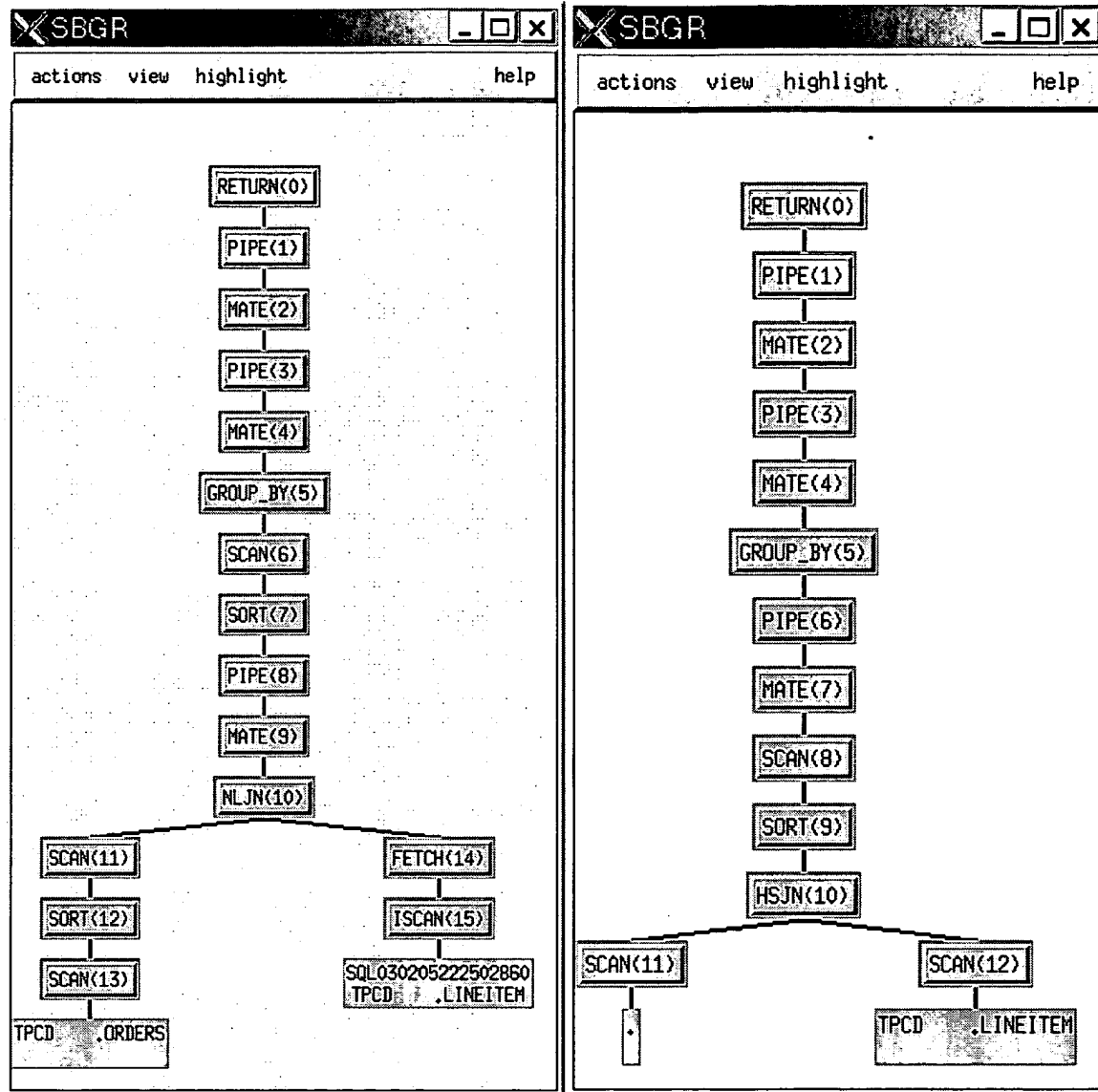


FIG. 14

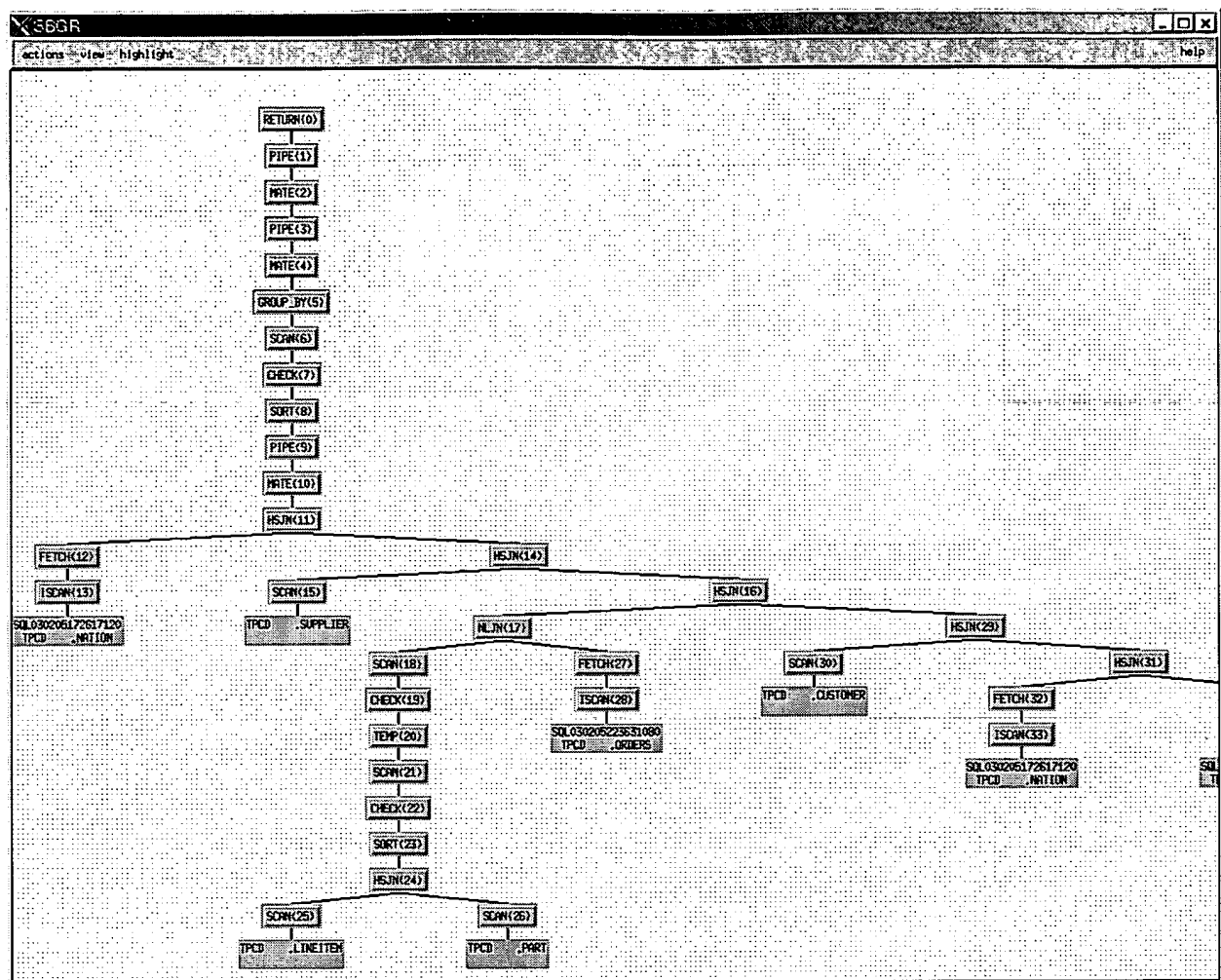


FIG. 15

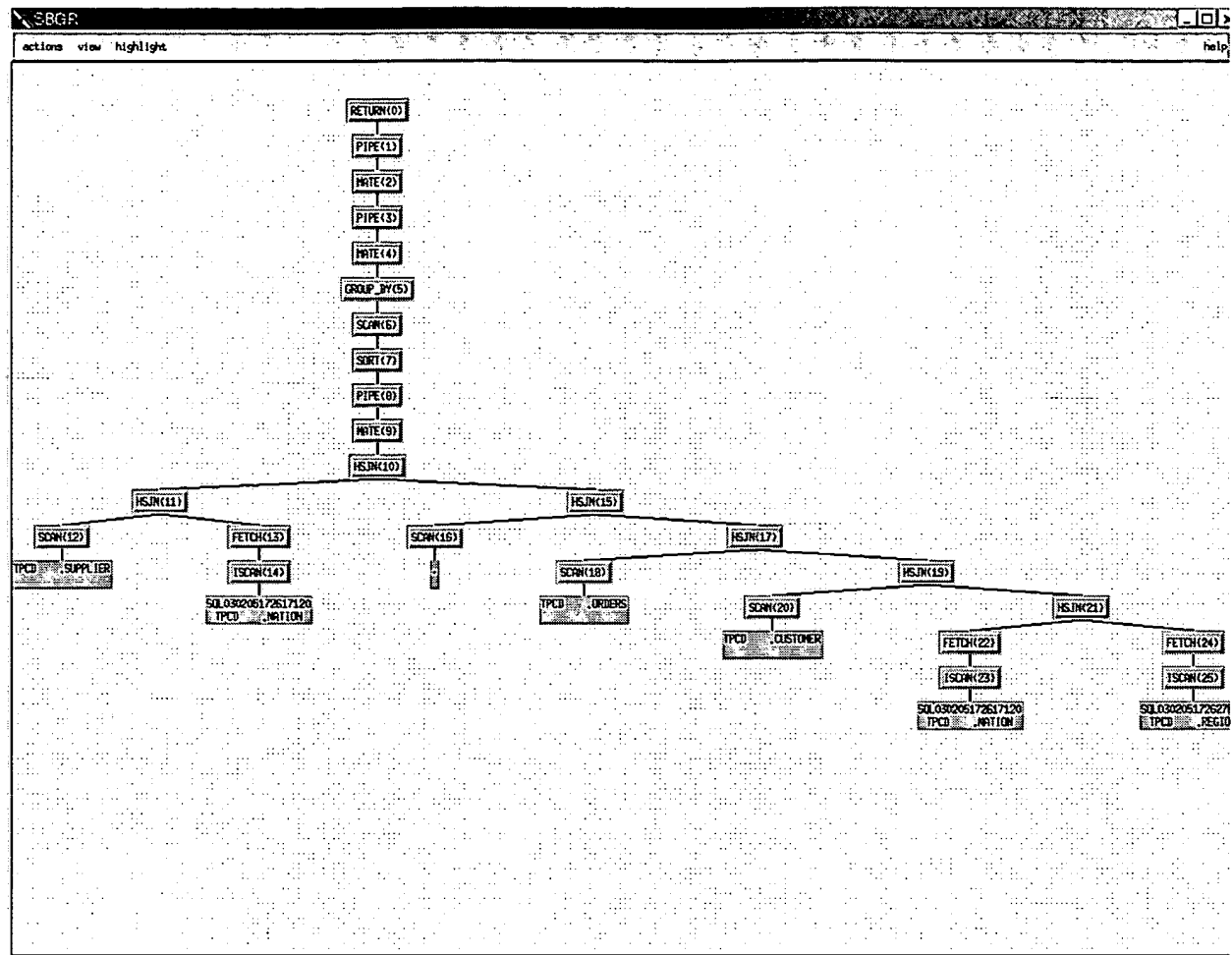


FIG. 16



**TABLE 1**

Checkpoint Type	Placement	Risk	Opportunity
Lazy Check (LC)	CHECK Above materialization points	Very Low -- only context switching	Low, only at materialization points
Lazy Check with Eager Materialization (LCEM)	CHECK-Materialization pairs on outer of NLJN	Context Switching + materialization overhead	Materialization points and NLJN outers
Eager Check with Buffering (ECB)	BUFCHECK on outer of NLJN.	Lower than LCEM till SafeEagerThreshold, gradually increases afterward	Can reoptimize anytime during materialization
Eager Check without compensation (ECWC)	CHECK below materialization points	High – may throw away arbitrary amount of work during reoptimization	Anywhere below a materialization point
Eager Check with deferred compensation (ECDC)	CHECK and INSERT before reoptimization; anti-join afterward	High – may throw away arbitrary amount of work during reoptimization	Anywhere in the plan.